Exact kinematics analysis of Car's suspension mechanisms using symbolic computation and interval analysis

Yves Papegay, Jean-Pierre Merlet, David Daney

Exact Kinematics Analysis of Car's Suspension Mechanisms Using Symbolic Computation and Interval Analysis

Yves A. Papegay [a,*] Jean-Pierre Merlet [a] David Daney [b]

[a] *INRIA – COPRIN, 2004 route des Lucioles, 06902 Sophia Antipolis, France*

[b] *LORIA – SPACES, Campus Scientifique, 54506 Vandoeuvre–lès–Nancy, France*

Abstract

This paper shows how some *exact computational methods* based on interval analysis have been used for the kinematic analysis of mechanisms involved in suspension of vehicles. It points out that, among the tools recently developed to solve exactly sets of equations, some are powerful and efficient enough to analyse and solve the kinematics of such mechanisms. This paper presents numerical results which are much more accurate and complete than the one obtained through numerical iterative computations. Those results are illustrated on two representative suspension mechanisms.

Key words: Interval analysis, Mechanisms, Kinematics, Symbolic Computation, System solving, Modelling, Simulation

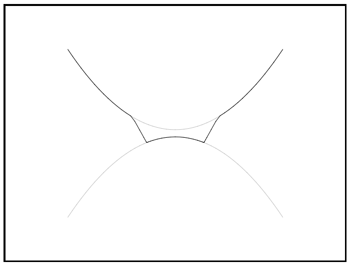
---

1 Introduction

Multi-link suspensions are yet very popular on a lot of vehicle and the numerical simulation of their motions is a critical step of their design process. Hence the kinematics of these mechanisms are the main topics of a large number of studies ([1], [2], [3]) even if mostly the simplest case of the well known Mac Pherson mechanism has been thoroughly investigated ([4],[5]).

Among the papers addressing the problem of the kinematics analysis of such mechanisms, not one (except [6]) considers the problem of the reliability of

* Corresponding Author.
Email address: yves.papegay@sophia.inria.fr (Yves A. Papegay).



Grayed curves are the two branches of the solution curve, which correspond to two different trajectories, black curve is the computed trajectory.

Fig. 1. a jump to alternative solution branches

the Newton-Raphson iterative scheme that is usually used to solve the highly non-linear equations appearing in the kinematics. Indeed, being given a set of numerical values for the parameters of the mechanism, such a numerical iterative process leads to a single result for the position of the wheel while the system of equations usually have several real solutions corresponding to several different positions. So, when such a process is used at each step for computing a discrete approximation of the trajectory of the wheel during a motion, there is no guarantee that the whole set of points defining the positions are belonging to the same branch of the corresponding curve, especially as the Newton-Raphson scheme is not guaranteed to converge toward the solution which is the closest to the initial guess (a fact which is unknown to many design engineers). Hence, the reliability of the computed trajectory is done by a visual check which can be faulty if two trajectories came close to each other at some point of the process. An exemple of obvious jumps between alternative solution branches is shown on figure 1 but such a jumps are not always easily detectible.

Small differences on camber or toe angle or on orientation of steering axis may have a drastical influence on the handling performances of a car either directly or through their rates of change with jounce-rebound. Hence, it is natural to investigate if existing computational methods can give more precise and complete qualitative results and/or more accurate quantitative results for the kinematics analysis of suspension mechanisms. For this challenging goal, *exact computational methods* are good candidates. It is a general terminology for any kind of method for solving equations which returns the whole set of real solutions, given with its accuracy. It includes symbolical methods based on algebraic properties of the equations, certified numerical methods based on analytical properties of the equations and hybrid methods combining a symbolic pre-processing of the equations and a certified numerical solving of the resulting problem.

In section 2, we will describe how we suggest to build the geometrical model of multi-link mechanisms for suspensions for their kinematics analysis and what is the algebraic structure of this model and we will briefly review some recent methods for computing exactly the solutions of systems of equations. In section 3, we will describe more precisely the method based on interval analysis, that we have extensively experimented. In sections 4 and 5, we will present detailed studies of the kinematics of two typical mechanisms – MacPherson Strut and 5-rod 5SK mechanism – which respectively corresponds to one of the simplest and one of the most complicated mechanisms and leads to typical systems of equations.

2 Modelling Suspension Mechanisms

At an early stage of the design process, a suspension mechanism is modelled as a multibody rigid system forming several closed kinematic chains in the three-dimensional space. Most of the joints between the bodies are physically elastokinematic joints but are modelled as spherical (ball) joints. Revolute joint may also appear, especially after simplifications due to an analysis of the geometry of the mechanism. Some advanced mechanisms also include cylindrical joints and more seldom universal joints.

2.1 Geometrical analysis

In the geometrical analysis, we express the constraints induced by the presence of a joint between two bodies by writing equations that imply that a point has to remain respectively on a cylinder, a circle or a sphere – from this point of view, it is not necessary to distinguish between spherical and universal joint.

This set of algebraic non-linear equations expressing the mechanical constraints is known as *geometrical model using natural coordinates*. Equations involve lengths and three-dimensional position of selected points as parameters, and are algebraic with respect to them. They are instances of one of the following formulas,

- for a cylindrical joint:

$$\begin{aligned} l_{ij}^2 = & (v_i^2 + w_i^2)(x_j - x_i)^2 + (u_i^2 + w_i^2)(y_j - y_i)^2 \\ & +(u_i^2 + v_i^2)(z_j - z_i)^2 - 2u_i v_i (x_j - x_i)(y_j - y_i) \\ & -2u_i w_i (z_j - z_i)(x_j - x_i) - 2v_i w_i (y_j - y_i)(z_j - z_i)) \end{aligned} \tag{1.1}$$

- for a revolute joint:

$$\begin{cases} (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 & = l_{ij}^2 \\ (x_i - x_j)u_i + (y_i - y_j)v_i + (z_i - z_j)w_i & = k_{ij} \end{cases} \tag{1.2}$$

- for a spherical or universal joint:

$$(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 = l_{ij}^2 \tag{1.3}$$

where (x_i, y_i, z_i) and (x_j, y_j, z_j) denotes the coordinates of two points belonging to each of the linked bodies, and where (u_i, v_i, w_i) and k_{ij}, l_{ij} denotes the coordinates of an unitary vector and other constants which are only depending on the geometry of the mechanism.

Depending of the choice of the points involved in the equations, the model may also contains equations expressing that n points are belonging to the same rigid body. It consists in $\frac{n(n-1)}{2}$ distance equations when $n \leq 4$. When $n \geq 5$ we first select 4 points, write the 6 corresponding distance equations and then we write the remaining points as combinations of this 4 points.

2.2 Kinematics Analysis

Several kinematics analysis can be performed with the same geometrical model according to which parameters are considered as known. If we address a design problem, a simulation or a calibration problem, some parameters will be variable and be therefore the unknowns of the problem while the other will be fixed.

That is why, once the geometrical analysis has been done, we have to separate parameters in three classes, physical constants, *input* variables, and *output* variables. The kinematics analysis consists in studying how to express output parameters in terms of the constants and the input parameters.

As most of the equations of the geometrical model are non-linear, there are usually, for a given set of values of the input variables, several sets of values of the output which are solutions of the system of equations.

The kinematics analysis, which aims at describing all the possible motions for the mechanism, has two parts:

- *Solving* i.e to find all the sets of values corresponding to real solutions for the output, being given a set of values of the input parameters. It has to be done a lot of time for a wide range of values of these parameters.
- *Following trajectories*, by describing how each solution varies when the input is changing continuously, with a special attention to pay at singular points, where these trajectories are crossing.

Suspension mechanisms are usually one degree of freedom mechanisms (two for front wheels). In most of the cases, the kinematics analysis consists in expressing the pose of the wheel as a function of the length of the spring.

3 Exact Computational Methods

Three important exact computational methods have been reviewed by Raghavan and Roth in [9] in the scope of kinematics analysis problems, *Gröbner basis*, *continuation* and *dialytic elimination*. They are respectively representative of the three family of methods, *symbolical* one, *numerical* one and *hybrid*.

In this paper we will focus on the two first one, symbolical methods because they return explicit symbolic expressions of the output parameters in terms of the input and fixed parameters and provides useful information about each branch of a trajectory through the computed expressions, and certified numerical methods based on interval analysis, because they allow to take in account the numerical uncertainty on the fixed parameters.

Symbolical methods like Gröbner basis or Triangular set generalize the Gauss elimination to set of non-linear equations. Unfortunately the number of monomials involved in intermediate expressions is huge even for academic examples and is very quickly increasing with the number of variables and the number of equations, as well as the size of the rational coefficients. Even if the theoretical complexity bound (twice exponential in the worst cases) is drastically reduced on practical example (simple exponential when solutions are isolated points), we will show that it is not practicable, even for the simple MacPherson mechanism, to compute the symbolical expression of the output parameters using such a method.

We do not consider is this paper hybrid method even if recent one are very efficient for solving set of polynomial equations, as the resultant based one ([10]), or the one implemented in GB and RS ([15],[16]) based on Gröbner basis. Those methods, like certified numerical one are "local" methods providing a numerical result for each given set of input parameters. Hence they have no capabilities for following trajectories. We have designed specialized numerical procedure for this purpose in the framework of certified numerical methods and, for a fair comparison we should have done the same work for hybrid method.

3.1 Solving with Interval Analysis

In method based on *interval analysis*[11], each variable X is replaced by a range $[\underline{x}, \overline{x}]$ and each mathematical operator has a new definition. For example the "+" operator for two ranges X_1, X_2 is defined as the range X_{12} such that:

$$X_{12} = [\underline{x_1} + \underline{x_2}, \overline{x_1} + \overline{x_2}]$$

which means that the sum of any number in the ranges X_1, X_2 is strictly included in the range X_{12}. As the operators that have an interval definition are almost all the classical mathematical functions, interval analysis enable to determine lower and upper bounds for the value of any mathematical expression. Clearly if 0 is not included between these bounds, then the value of the expression will never be 0 whatever the values of the unknowns. A drawback is that usually the bounds will be overestimated. For example if we consider the function $F = x^2 + x$ when $x \in [-1, 3]$ then:

$$F([-1, 3]) = [0, 9] + [-1, 3] = [-1, 12]$$

while the real bounds are [-0.25,12]. But an advantage of interval computation is that it is possible to implement interval arithmetics in such a way that round-off errors are taken into account i.e. the interval evaluation of a function is guaranteed to include the real bounds for the function.

A basic equation solver based on interval analysis will use a *bisection process* that we will illustrate on the equation F in one unknown. If we are looking for the solution of $F(x) = 0$ when x lie in the range [-1,3] we will start by computing the interval evaluation of the function for this range. As this lead to the range [-1,12] we cannot conclude if F has a real root in the range. The bisection process consists in splitting the initial interval into two new intervals, for example [-1,1],[1,3] and repeat the process on each new interval. In our case the interval evaluation of F for the range [1,3] leads to [2,12] that does not include 0. Hence this interval will be rejected and we will repeat the bisection process only on the interval [-1,1]. The process will return as solution the range whose width (the difference between the upper and lower bound) is lower than a given threshold and for which the interval evaluation include 0.

Note that this basic solver will not be efficient in practice if it is not completed by filtering operators (that reduce the width of the interval without performing a bissection [13]), existence operators (that determine there is a unique solution within the current ranges and allow to compute numerically this root [11], [12]).

It exists general purposes filtering and existence operators, i.e. operators that works for any equations. But it is also possible to develop specific operators for specific classes of equations, that are much more efficient then the general purpose version. In ALIAS – a C++ library [1] that implement high-level solving procedures using the `BIAS/Profil` library [2] for the basic interval operations [19] – specific operators have been developped for distance equations, that are common in mechanism theory, and the results presented in this paper have

[1] `www-sop.inria.fr/coprin/software/ALIAS/`
[2] `www.ti3.tu-harburg.de/Software/PROFILEnglisch.html`

been obtained with this solver.

Interval analysis may also be used if the coefficients of the equations have not afixed value but are interval (e.g. if they result from a measurement on the physical system). In that case, solutions will be represented as ranges for the unknowns. It is guaranted that *all* solutions will lie in the ranges but not all the proposed ranges may contain a solution.

3.2 Following a trajectory

As we have seen there are various tools to determine all the possible configurations of the mechanism for a given value of its free parameter l. But what we are really interested in is to determine the configurations for several values of this free parameter, say at l_0, $l_0 + \Delta l$, $l_0 + 2\Delta l$, $\ldots l_1$ where Δl is an arbitrary step size, i.e. to get an approximation of the trajectories of the mechanism as function of the degree of freedom. We may evidently use repeatedly the solving procedures at each step but this will lead to a lengthy procedure and furthermore problems may occur if two trajectories come close to each other as we may have difficulties to determine to which branch belong a configuration. Alternatively we may want to follow a given branch being given an approximate value for the configuration of the mechanism.

The usual approach to both problems is to use the result of the solving procedure at one step as the initial guess of a Newton scheme that will be used to determine the configuration for the next step. But this is an unsafe method as there is no guarantee that the Newton scheme will converge to the solution that belong to the same branch. We propose to improve this method by using Kantorovitch theorem. Let a system of n equations in n unknowns:

$$F = \{F_i(x_1, \ldots, x_n) = 0, i \in [1, n]\}$$

Let $\mathbf{x_0}$ be a point, called the *central point*, and $U = \{\mathbf{x}/||\mathbf{x} - \mathbf{x_0}|| \leq 2B_0\}$, the norm being $||A|| = \text{Max}_i \sum_j |a_{ij}|$. Assume that $\mathbf{x_0}$ is such that:

(1) the Jacobian matrix of the system has an inverse Γ_0 at $\mathbf{x_0}$ such that $||\Gamma_0|| \leq A_0$
(2) $||\Gamma_0 F(\mathbf{x_0})|| \leq 2B_0$
(3) $\sum_{k=1}^{n} |\frac{\partial^2 F_i(\mathbf{x})}{\partial x_j \partial x_k}| \leq C$ for $i, j = 1, \ldots, n$ and $\mathbf{x} \in U$
(4) the constants A_0, B_0, C satisfy $2nA_0B_0C \leq 1$

Then there is an unique solution of $F = 0$ in U and Newton scheme used with any $\mathbf{x}$ in U as estimate of the solution will converge toward this solution [12].

Let us assume that we have computed the set of m solutions of the system F, $\{X_1 = (x_1^1, \dots x_n^1), \dots X_m = (x_1^m, \dots x_n^m)\}$, for a given value l_0 of the degree of freedom l. Let d_j be the minimal distance between the solutions set X_j and all the other solutions. Assume now that the degree of freedom is changed to $l_0 + \Delta l$. We use Kantorovitch theorem for each solution set X_j with as central point the set itself. Two cases may occur when applying this theorem on the set X_j:

(1) the hypothesis of Kantorovitch theorem are satisfied and $d_j > 2B_0$
(2) the hypothesis of Kantorovitch theorem are not satisfied or $d_j < 2B_0$

In the former case Newton scheme applied to the kinematic equations obtained for $l_0 + \Delta l$ with as initial guess X_j will converge to a solution that lie on the same branch than X_j (otherwise there will be two solutions within the ball, which contradict Kantorovitch theorem). In the later case we are not sure that Newton scheme will converge: hence we will consider the system of equations obtained for $l_0 + \Delta l/2$ and repeat the procedure. Here also two cases may occur:

(1) the process is successful for a given value of the degree of freedom
(2) the process is not successful even for a very small change to l

In the former case we will apply the procedure to another branch and a synchronisation mechanism will ensure that each branch is followed in sequence and that the configuration solution of each branch are computed for each Δl step. In the later case Kantorovitch will fail because the central point of a branch is too close to a configuration in which the jacobian of the system is degenerate, i.e. we are close to a singularity of a mechanism and at least two branches collapse. In that case the procedure will signal that a singularity has been encountered. As we need a new starting point to follow the branches we will start again a solving procedure on the system obtained when the degree of freedom is $l_0 + \Delta l$. If the procedure is unable to find a Kantorovitch solution with this new value we change it to $l_0 + 2\Delta l$ and so on until a valid set of solutions is found.

A drawback of this method is that we will follow only the branches that have been initially detected when using the solving procedure: isolated branches that may appear for a value of the degree of freedom strictly included between l_0 and l_1 will not be detected (although we will detect if these branches have an intersection with the known branches). This is not a problem if we are interested in following only one branch, as this is usually the case for the suspension mechanisms.

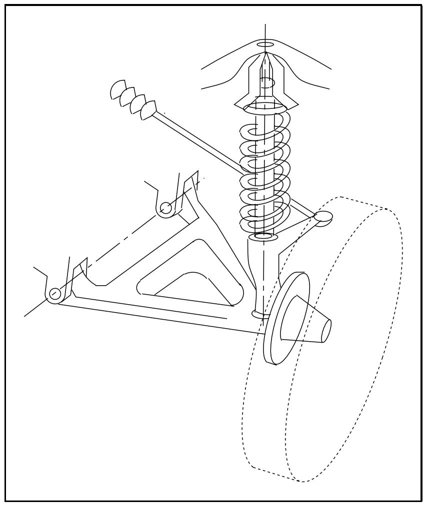


Fig. 2. MacPherson mechanism

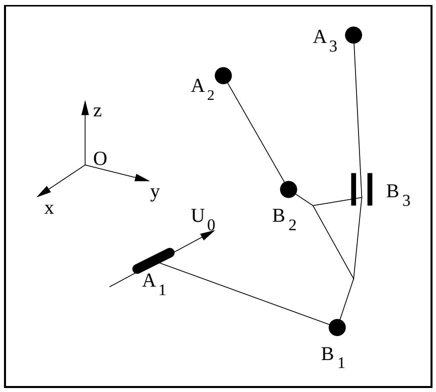


Fig. 3. MacPherson - notations

4 MacPherson mechanism

The MacPherson[3] suspension mechanism [5] - shown on figure 2 - is the most widely employed front suspension mechanism for small cars. It is made of three main components linking the vehicle body to the wheel spindle,

(1) the lower arm, attached to the body through a revolute joint and to the spindle through a spherical joint,
(2) the tie rod, for steering control, which is a spherical-spherical crank,
(3) and the strut of variable length rigidly fixed on the wheel side and fixed with a ball joint on the body side – a spring-damper system connect the ball joint and the wheel spindle.

This mechanism has only one active degree of freedom corresponding to the length of the strut.

[3] from the name of its designer in the 1940s, Earl S. MacPherson, General Motors

4.1 Geometrical analysis

For the geometrical analysis, we assume the notations given in figure 3. We define a point at each extremity of each link, and we denote by (u_i, v_i, w_i) – resp. (x_i, y_i, z_i) – the coordinates of point A_i – resp. B_i – relative to a fixed reference frame attached to the body of the car (O, Ox, Oy, Oz). Moreover we define a point B_2 at the other end of the coiled spring. We represent by l_i the distance between point A_i and point B_i, by l_{ij} the distance between point B_i and point B_j, and U_0 is a unit vector of coordinates (u_0, v_0, w_0) along the rotation axis of the lower arm. We also need to introduce two constants, k_{13} and k_{23} describing respectively the angles (B_3B_1, B_3A_3) and (B_3B_2, B_3A_3).

As the mechanism has one degree of freedom, the purpose of the kinematic analysis is to determine the pose of the wheel according to the value of one variable parameter of the system. The pose of a body is fully defined if the coordinates of three points on it are known in the reference frame – here, we choose B_3, B_4 and B_5. We select as the variable parameter the length of the spring, l_3.

Using the geometrical model defined in section 2.1, expressing the constraints for each link leads to the following system of equations:

$$(u_1 - x_1)u_0 + (v_1 - y_1)v_0 + (w_1 - z_1)w_0 = 0 \tag{2.1}$$
$$(x_1 - u_1)^2 + (y_1 - v_1)^2 + (z_1 - w_1)^2 = {l_1}^2 \tag{2.2}$$
$$(x_2 - u_2)^2 + (y_2 - v_2)^2 + (z_2 - w_2)^2 = {l_2}^2 \tag{2.3}$$
$$(x_2 - u_3)^2 + (y_2 - v_3)^2 + (z_2 - w_3)^2 = {l_3}^2 + 2k_{23}l_3l_{23} + {l_{23}}^2 \tag{2.4}$$
$$(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 = {l_{12}}^2 \tag{2.5}$$
$$(x_1 - u_3)^2 + (y_1 - v_3)^2 + (z_1 - w_3)^2 = {l_3}^2 + 2k_{13}l_3l_{13} + {l_{13}}^2 \tag{2.6}$$
$$(x_5 - u_3)^2 + (y_5 - v_3)^2 + (z_5 - w_3)^2 = {l_3}^2 \tag{2.7}$$
$$(x_5 - x_1)^2 + (y_5 - y_1)^2 + (z_5 - z_1)^2 = {l_{13}}^2 \tag{2.8}$$
$$(x_5 - x_2)^2 + (y_5 - y_2)^2 + (z_5 - z_2)^2 = {l_{23}}^2 \tag{2.9}$$

As A's points are fixed points, we only have as unknown variables the 9 coordinates of B_3, B_4 and B_5. Other symbols correspond to fixed (but not independent) parameters, except one of them taken as input, say for example l_3 expressing the degree of freedom.

4.2 Solving by Symbolical Methods

It is completely hopeless to enter this set of equations in a computer algebra system like MAPLE[14] and to try to solve it symbolically with the help of the dedicated algebraic solving function to get the symbolic expressions of the 9 variables in term of the parameters, especially without any pre-processing of the equations.

Using a specialised software like MACAULAY[17] designed for solving very efficiently algebraic systems of equations, we can have information on the set of solutions. In this case, we get that the system is algebraic set of dimension 20 and degree 1024, which means that the values of 20 parameters may be arbitrary fixed and once they are fixed there are at most 1024 set of solutions for the 9 unknowns. Theoretically, we may also obtain a symbolic expression of the solutions by computing a Gröbner basis for the right elimination order on the variables. But, due to the size of the problem and the high complexity of the method, we did not succeed in getting the first result: MACAULAY fails for memory size (exceeding 968 Megabytes) after having computed during 456 hours of CPU time on a DEC ALPHA 1 with 633 Mb of RAM.

The next challenge is to get the expressions of the 9 variables, not as a whole symbolic formula, but as an evaluation program, i.e. a sequence of symbolic formulas, each of them expressing a new parameter in terms of the parameters previously expressed, for example

$$x_1 = f_1(s, l_3) \tag{3.1}$$
$$y_1 = f_2(s, l_3, x_1) \tag{3.2}$$
$$z_1 = f_3(s, l_3, x_1, y_1) \tag{3.3}$$
$$z_2 = f_4(t, l_3, x_1, y_1, z_1) \tag{3.4}$$
$$x_2 = f_5(t, l_3, x_1, y_1, z_1, z_2) \tag{3.5}$$
$$y_2 = f_6(t, l_3, x_1, y_1, z_1, z_2) \tag{3.6}$$
$$z_5 = f_7(t, l_3, x_1, y_1, z_1, x_2, y_2, z_2) \tag{3.7}$$
$$x_5 = f_8(t, l_3, x_1, y_1, z_1, x_2, y_2, z_2, z_5) \tag{3.8}$$
$$y_5 = f_9(t, l_3, x_1, y_1, z_1, x_2, y_2, z_2, z_5) \tag{3.9}$$

where s and t denote respectively $u_1, v_1, w_1, u_0, v_0, w_0, l_1, u_3, v_3, w_3, k_{13}, l_{13}$ and $s, u_2, v_2, w_2, l_2, k_{23}, l_{23}, l_{12}$.

On our system, it can be achieved with a computer algebra system as MAPLE by using the following algorithm:

(1) Solve equations (2.1) to get the expression of z_1 in term of x_1 and y_1,

$$z_1 = \frac{-u_0 x_1 + u_0 u_1 - v_0 y_1 + v_0 v_1 + w_1 w_0}{w_0} \tag{3.10}$$

(2) substitute the previous result into (2.2) to eliminate z_1 from the equation and solve it to get the expression of y_1 in term of x_1, we get two

expressions

$$y_1 = 1/2 \frac{2{w_0}^2 v_1 - 2u_0 x_1 v_0 + 2{v_0}^2 v_1 + 2u_0 u_1 v_0 \pm 2\sqrt{\Delta}}{{v_0}^2 + {w_0}^2} \tag{3.11}$$

with

$$\begin{aligned}\Delta = & 2{v_0}^2{w_0}^2 x_1 u_1 + 2{w_0}^2{u_0}^2 x_1 u_1 - {w_0}^4{x_1}^2 - {w_0}^4{u_1}^2 + {w_0}^4{l_1}^2 \\ & +2{w_0}^4 x_1 u_1 - {w_0}^2{u_0}^2{x_1}^2 - {w_0}^2{u_0}^2{u_1}^2 - {v_0}^2{w_0}^2{x_1}^2 - {v_0}^2{w_0}^2{u_1}^2 \\ & +{v_0}^2{l_1}^2{w_0}^2\end{aligned}$$

Note that these expressions correspond to real values of y_1 only when x_1 is such as Δ is non-negative.

(3) substitute (3.10) and one (resp. the other) of the two previous results (3.11) into (2.6) to eliminate both y_1 and z_1 and obtain a quadratic equation governing x_1, solve it (in both cases) to get (in each case) two results expressing x_1 in terms of the fixed parameters and l_3. We obtain four results on the same pattern:

$$x_1 = \frac{4\sqrt{P_i(t)} + Q_i(t)}{R_i(t)} \tag{3.12}$$

where $R_i(t)$ is a polynomial with 30 monomials of degree 4, $Q_i(t)$ is a polynomial with 106 monomials of degree 5 or 6 and $P_i(t)$ is a polynomial with 2047 monomials of degree 10, 11 or 12. They are as many candidates for the f_1 expression of the program, but only two at most are valid one.

(4) Compute the differences between equation (2.5) and equation (2.4) and between equation (2.5) and equation (2.3). Quadratic terms in x_2, y_2 or z_2 are vanishing and we get two linear equations in those variables:

$$\begin{aligned}&(-2x_1 + 2u_3)x_2 + (-2y_1 + 2v_3)y_2 + (-2z_1 + 2w_3)z_2 \\ &\quad -{v_3}^2 + {x_1}^2 - {u_3}^2 + {y_1}^2 + {z_1}^2 - {w_3}^2 \\ &\qquad = {l_{12}}^2 - {l_3}^2 - 2k_{23} l_3 l_{23} - {l_{23}}^2\end{aligned} \tag{3.13}$$

$$\begin{aligned}&(-2x_1 + 2u_2)x_2 + (-2y_1 + 2v_2)y_2 + (-2z_1 + 2w_2)z_2 \\ &\quad -{v_2}^2 + {x_1}^2 - {u_2}^2 + {y_1}^2 + {z_1}^2 - {w_2}^2 \\ &\qquad = {l_{12}}^2 - {l_2}^2\end{aligned} \tag{3.14}$$

By solving them, express x_2 and y_2 in terms of z_2:

$$\begin{aligned}-2Dy_2 = & (-2u_2 z_1 + 2x_1 w_2 - 2x_1 w_3 + 2u_2 w_3 - 2u_3 w_2 + 2u_3 z_1)z_2 - x_1{l_3}^2 \\ & -x_1{l_{23}}^2 - 2x_1 k_{23} l_3 l_{23} + 2u_2 k_{23} l_3 l_{23} + x_1{l_2}^2 + x_1{w_3}^2 + x_1{u_3}^2\end{aligned}$$

$$+{x_1}{v_3}^2-{u_2}{v_3}^2-{u_2}{w_3}^2-{u_2}{l_{12}}^2+{u_2}{l_3}^2+{u_2}{l_{23}}^2-{x_1}{u_2}^2 \tag{3.15}$$
$$+{u_2}{z_1}^2-{u_2}{u_3}^2+{u_3}{v_2}^2+{u_3}{u_2}^2-{u_3}{z_1}^2-{u_3}{y_1}^2-{u_3}{x_1}^2$$
$$-{u_3}{l_2}^2-{x_1}{w_2}^2-{x_1}{v_2}^2+{u_2}{y_1}^2+{u_3}{l_{12}}^2+{u_3}{w_2}^2+{u_2}{x_1}^2$$

$$\begin{aligned}2Dx_2 = {} & (-2w_2v_3+2z_1v_3-2y_1w_3+2v_2w_3+2w_2y_1-2v_2z_1)z_2 \\ & -2y_1k_{23}l_3l_{23}+2v_2k_{23}l_3l_{23}+y_1{w_3}^2+y_1{u_3}^2-{z_1}^2v_3+{l_{12}}^2v_3 \\ & -{v_2}^2y_1+{v_2}^2v_3-{w_2}^2y_1-{l_2}^2v_3-{x_1}^2v_3+v_2{y_1}^2+v_2{x_1}^2 \qquad (3.16) \\ & +{l_2}^2y_1-v_2{u_3}^2-v_2{v_3}^2-v_2{w_3}^2-v_2{l_{12}}^2+v_2{l_3}^2+v_2{l_{23}}^2 \\ & +v_2{z_1}^2+y_1{v_3}^2+{u_2}^2v_3-{u_2}^2y_1+{w_2}^2v_3-{y_1}^2v_3-y_1{l_{23}}^2-y_1{l_3}^2\end{aligned}$$

with

$$D=-u_2y_1-u_3v_2+u_3y_1+x_1v_2-x_1v_3+u_2v_3$$

(5) substitute the two previous expressions in equations (2.3) to eliminate both x_2 and y_2, solve it to get the expression of z_2 in terms the fixed parameters, l_3 and x_1, y_1, z_1. We obtain two results on the same pattern:

$$z_2=\frac{4\sqrt{P_i(t,x_1,y_1,z_1)}+Q_i(t,x_1,y_1,z_1)}{R_i(t,x_1,y_1,z_1)} \tag{3.17}$$

where $R_i(t,x_1,y_1,z_1)$ is a polynomial with 63 monomials of degree 4, where $Q_i(t,x_1,y_1,z_1)$ is a polynomial with 305 monomials of degree 5 or 6 and $P_i(t,x_1,y_1,z_1)$ is a polynomial with 14275 monomials of degree 10, 11 or 12.

(6) execute analogous steps to step 4 and 5, replacing respectively equations (2.3), (2.4) and (2.5) by (2.7), (2.8) and (2.9) and the variables x_2, y_2, z_2 by x_5, y_5, z_5.

The implementation of this algorithm computes all the expressions appearing in the complete program in less then 2 minutes. And, if we give numerical values to the parameters – a realistic set of data is described below, unit is millimeter – we can use this evaluation program for computing numerically the solutions. This can be done in around 1 minute by MAPLE, but such a performance may be largely enhanced by generating a C program and using it for the numerical computations.

$$\begin{cases} u_0=0.9901559, v_0=0.09522448, w_0=-0.1025845607, u_1=33.69279, v_1=342.8049, \\ w_1=-0.03617776374, l_1=317.4905466, l_2=306.8979231, l_{12}=180.2625676, l_{13}=126.3992495, \\ k_{23}=-0.02052180122, k_{13}=0.86574862, l_{23}=134.0027263, u_2=106, v_2=290, \\ w_2=90, u_3=10, v_3=510, w_3=583 \end{cases}$$

The figure 4 and 5 presents respectively a 3-dimensional view of the set of acceptable positions for B_4 and for B_5 when the parameters l_3 varies between 231mm and 811mm. The figures 6 and 7 show the two curves obtained by

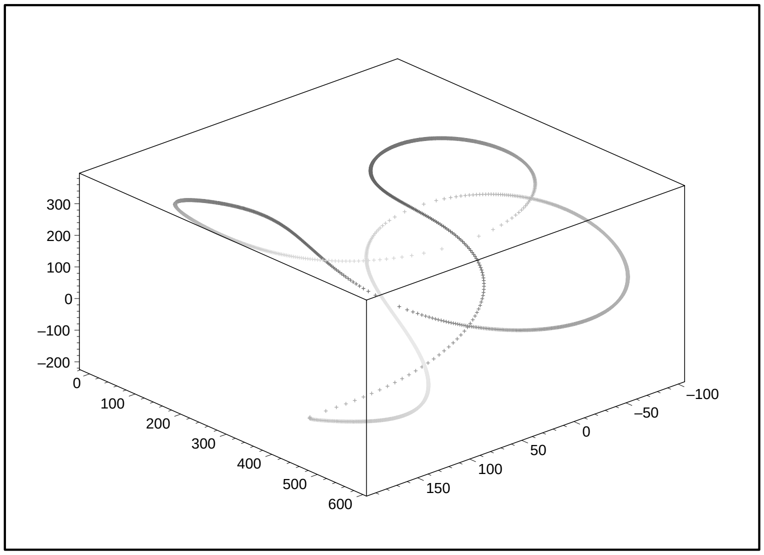


Fig. 4. acceptable positions for B_4

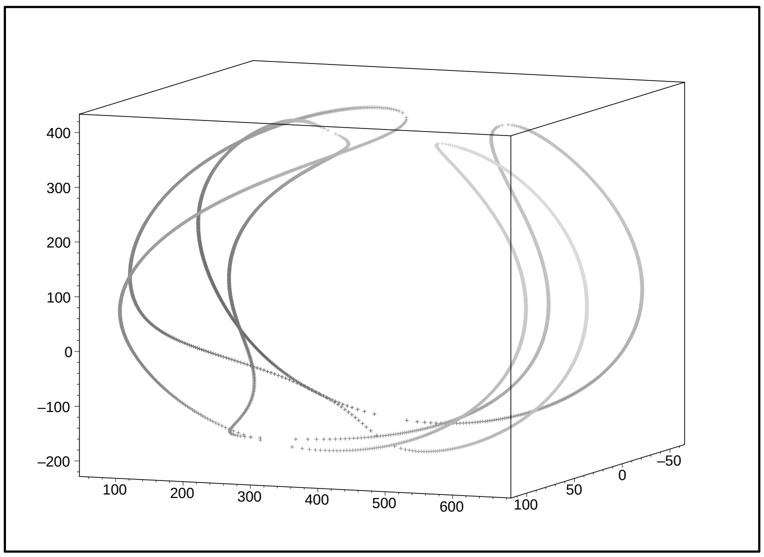


Fig. 5. acceptable positions for B_5

connecting the points of the set of acceptable positions for B_5, which are two separate supports of any trajectory of B_5.

4.3 Solving with ALIAS

Using the numerical data presented in the previous section and the algorithm presented in section 3.1 it is possible to obtain similar plots. On a Pentium III at 600 Mhz, the computation time for finding the 8 initial starting points of the branches is about 22 seconds while following the 8 branches takes about 5 seconds (0.6 second if only one branch is followed).

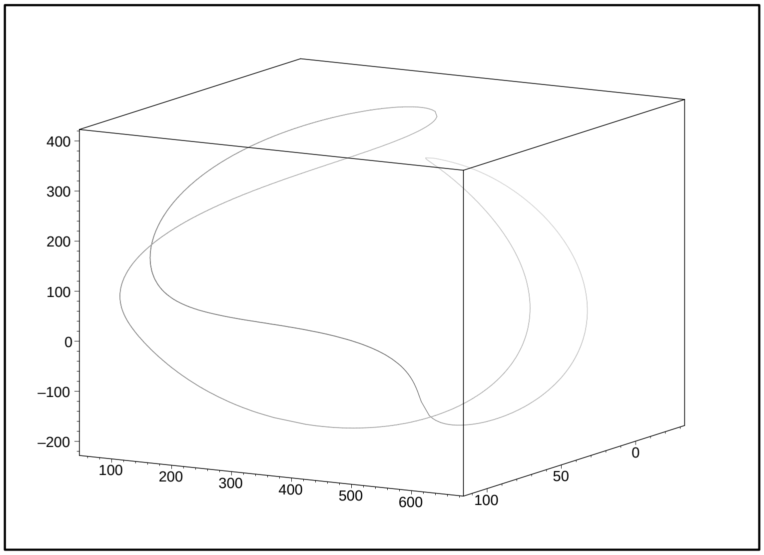


Fig. 6. acceptable positions for B_5 – first branch

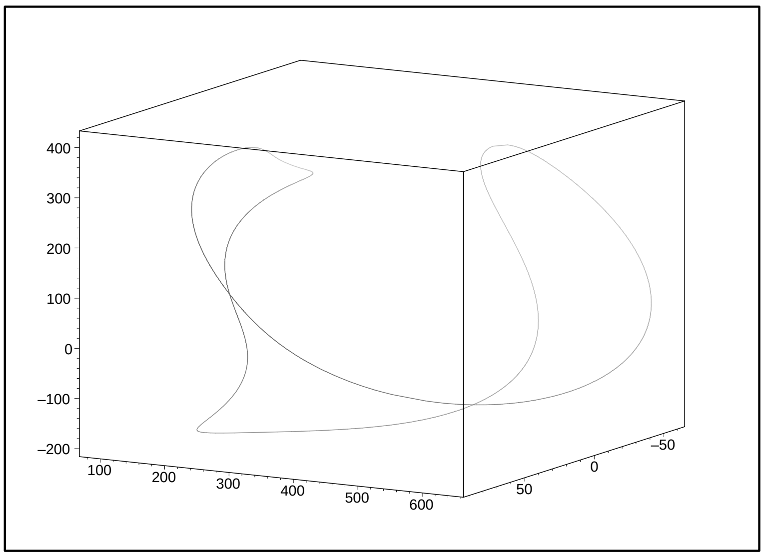


Fig. 7. acceptable positions for B_5 – second branch

5 5-rod 5SK

The mechanism, shown in figure 8, which is denoted by *5SK* in the standard classification[18] has been used for the design of the suspension of the rear wheels for more then 20 years on the Mercedes 190 series. The wheel support is linked to the body of the car through 5 rigid rods of fixed length. On both sides, the links between the rods and the other parts are realised by elasto-kinematic joints, modelled as ball joints. The spring-damper system (not shown on the figure) is connected to one of the 5 rigid rods and on the other side to the body of the car through ball joints.

From a kinematical point of view, this mechanism is highly redundant, it is

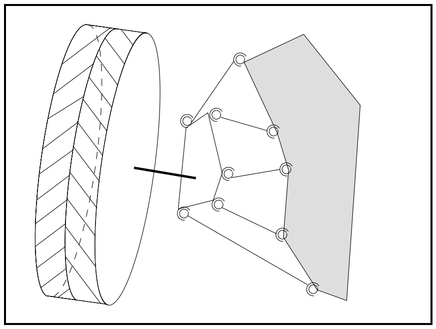


Fig. 8. 5-rod 5SK mechanism

made of 8 bodies – wheel support, 5 rods, car body and spring-damper – with 12 joints building 6 kinematics chains and 11 loops.

5.1 Geometrical analysis

The geometrical analysis is performed automatically as described in the section 2.1. The points on the body of the car are denoted by $A_i = (u_i, v_i, w_i)$ for $i = 1 \ldots 6$ and the points on the wheel support by $B_i = (x_i, y_i, z_i)$ for $i = 1 \ldots 5$ assuming that for each i, A_i is connected to B_i through the i-th rod. A_6 is the point where is attached on the body the spring-damper system and $B_6 = (x_6, y_6, z_6)$ is at the other end of this system, on the first rod. Let l_6 denotes the length of the spring damper system while the other parameters are physical constants describing the geometry of the mechanical system.

With these notations, constraints on the length of the rods can be written:

$$(u_i - x_i)^2 + (v_i - y_i)^2 + (w_i - z_i)^2 = l_i{}^2 \quad \text{for} \quad i = 1 \ldots 6 \tag{4.1}$$

We express the fact that the spring-damper system is attached to the first rod by:

$$\begin{cases} x_6 = k_6 u_1 + (1 - k_6) x_1 \\ y_6 = k_6 v_1 + (1 - k_6) y_1 \\ z_6 = k_6 w_1 + (1 - k_6) z_1 \end{cases} \tag{4.2}$$

And we express that B_1, B_2, B_3, B_4 and B_5 belongs to the same rigid body by:

$$(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2 = l_{ij}^2 \quad \text{for} \quad i, j = 1 \ldots 3 (i \neq j) \tag{4.3}$$

and by:

$$\begin{cases} x_4 = k_{14}x_1 + k_{24}x_2 + k_{34}x_3 \\ y_4 = k_{14}y_1 + k_{24}y_2 + k_{34}y_3 \\ z_4 = k_{14}z_1 + k_{24}z_2 + k_{34}z_3 \end{cases} \quad \text{and} \quad \begin{cases} x_5 = k_{15}x_1 + k_{25}x_2 + k_{35}x_3 \\ y_5 = k_{15}y_1 + k_{25}y_2 + k_{35}y_3 \\ z_5 = k_{15}z_1 + k_{25}z_2 + k_{35}z_3 \end{cases} \tag{4.4}$$

Altogether, these 18 equations build a redundant geometrical model of this mechanism governing the 18 coordinates of the points B_i. As only one rigid body may be moving in this mechanism, we know that theoretically, only 6 parameters are required to fully express its geometry. So, it is natural to try to reduce this model to a simpler one before going to the solving step.

By substitution of the expressions of the coordinates of B_4, B_5 and B_6 we get a reduced model with 9 equations and 9 unknowns:

$$(u_1 - x_1)^2 + (v_1 - y_1)^2 + (w_1 - z_1)^2 = {l_1}^2 \tag{5.1}$$

$$(u_2 - x_2)^2 + (v_2 - y_2)^2 + (w_2 - z_2)^2 = {l_2}^2 \tag{5.2}$$

$$(u_3 - x_3)^2 + (v_3 - y_3)^2 + (w_3 - z_3)^2 = {l_3}^2 \tag{5.3}$$

$$\begin{gathered}(u_4 - (k_{14}x_1 + k_{24}x_2 + k_{34}x_3))^2 + (v_4 - (k_{14}y_1 + k_{24}y_2 + k_{34}y_3))^2 + \\ (w_4 - (k_{14}z_1 + k_{24}z_2 + k_{34}z_3))^2 = {l_4}^2\end{gathered} \tag{5.4}$$

$$\begin{gathered}(u_5 - (k_{15}x_1 + k_{25}x_2 + k_{35}x_3))^2 + (v_5 - (k_{15}y_1 + k_{25}y_2 + k_{35}y_3))^2 + \\ (w_5 - (k_{15}z_1 + k_{25}z_2 + k_{35}z_3))^2 = {l_5}^2\end{gathered} \tag{5.5}$$

$$\begin{gathered}(u_6 - (k_6u_1 + (1 - k_6)x_1)))^2 + (v_6 - (k_6v_1 + (1 - k_6)y_1)))^2 + \\ (w_6 - (k_6w_1 + (1 - k_6)z_1)))^2 = {l_6}^2\end{gathered} \tag{5.6}$$

$$(u_2 - u_1)^2 + (v_2 - v_1)^2 + (w_2 - w_1)^2 = l_{12}^2 \tag{5.7}$$

$$(u_3 - u_1)^2 + (v_3 - v_1)^2 + (w_3 - w_1)^2 = l_{13}^2 \tag{5.8}$$

$$(u_3 - u_2)^2 + (v_3 - v_2)^2 + (w_3 - w_2)^2 = l_{23}^2 \tag{5.9}$$

5.2 Solving the kinematics analysis

We have made many attempt to get the symbolical expression of the B_i points in term of the constants, the positions of the A_i and the length l_6 of the spring-damper system. But, due to the huge size of the expression generated by the resolution, we didn't succeed even allowing a MAPLE process to use 10 Gigabytes of memory. We failed also in trying to get an evaluation program for computing the position of the B_i, as we did with the MacPherson mechanism.

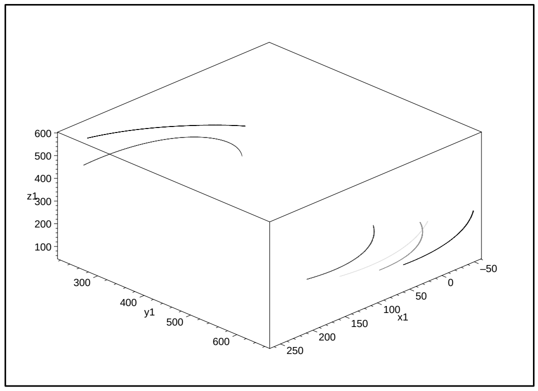


Fig. 9. The 6 trajectories of (x_1, y_1, z_1) when l_6 is in range [173,280]

In order to reduce the complexity of the computation, we gave explicit numerical values to some of the parameters using the following realistic set of data – unit is again millimeter.

$$\begin{cases} u_1 = -102.6, v_1 = 220.1, w_1 = 238.1, u_2 = 309.8, v_2 = 506, w_2 = 253.5, u_3 = 201.9, v_3 = 426.1, \\ w_3 = 294.1, u_4 = 206.7, v_4 = 443, w_4 = 398.2, u_5 = -2.1, v_5 = 363, w_5 = 421.5, u_6 = -112.5, \\ v_6 = 438.0, w_6 = 417.5, l_1 = 444.3865097, l_2 = 313.9128701, l_3 = 245.2243258, l_4 = 246.4982353, \\ l_5 = 297.6022345, l_{12} = 98.51543026, l_{13} = 191.7244116, l_{23} = 141.0111343, k_{14} = 1.706216400, \\ k_{24} = -2.641106193, k_{34} = 1.805302996, k_{15} = 2.559324080, k_{25} = -3.529570293, \\ k_{35} = 1.815276435, k_6 = 0.5 \end{cases}$$

Still with this numerical data we were not able to solve the problem using only symbolic computation. In that case however we may still solve the problem using the interval analysis approach. As mentioned in section 3.1 the first step of the approach is to determine the starting points of the branches by solving the system of equations for the initial value of the parameter.

5.2.1 Numerical example

Using the distance equations solver the computation time for determining the initial starting points for the branches is low. For example if we use the numerical data presented in the previous section the 6 initial points for $l_6 = 280$ are determined in about 50 seconds. Then, using the trajectory following algorithm, we are able to draw in an exact way the 6 trajectories obtained when l_6 varies between 174 and 280 in about 10 seconds. These trajectories are represented in figure 9, 10, 11.

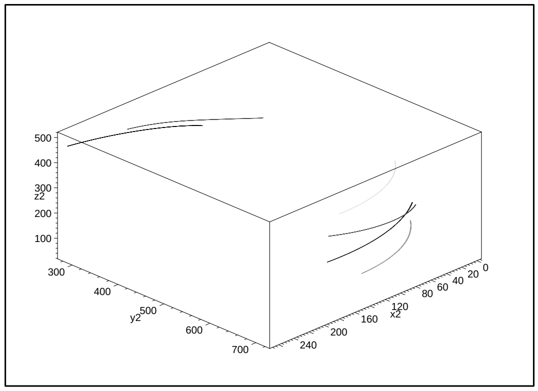


Fig. 10. The 6 trajectories of (x_2, y_2, z_2) when l_6 is in range [173,280]

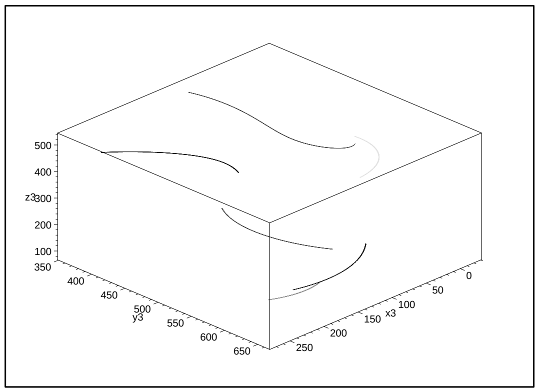


Fig. 11. The 6 trajectories of (x_3, y_3, z_3) when l_6 is in range [173,280]

6 Conclusion

We studied the kinematics of two typical suspension mechanisms, using two opposite approach to solve it exactly. The symbolic approach has the advantages to provide the kinematics solution directly as symbolic functions of the design parameters and is hence appropriate for design purposes but has very high costs and is not practicable for the kinematics of complex mechanisms. The interval-based approach that allows to solve the kinematics of all the suspension mechanisms but only for fixed values of their design parameters. It must be noted that both methods are exact in the sense that the computed trajectories are guaranteed. This is a large improvement as undetected jumps between branches may occur when using the usual method based on an iterative use of Newton scheme.

References

[1] M. Hiller and S. Frik. Five-link suspension. *Vehicle System Dynamics*, pages 254–262, 1993.

[2] M. Gadola, D. Vetturi, L. Manzo, and D. Cambiaghi. Multi-link suspension analysis: an elastokinematic model. In *Proceedings of the 29th International Symposium on Automotive Technology and Automation*, Florence, 1996.

[3] J. Knapczyk and S. Dzierzek. Elasto analysis of the five rod suspension considering join elasticties. In *Proceedings of the 9th World Congress on the Theory of Machines and Mechanisms*, pages 2824–2828, Milan, 1995.

[4] D.L. Cronin. Macpherson strut kinematics. *Mechanism and Machine Theory*, 16(5):631–644, 1981.

[5] C.H. Suh. Modeling the mcpherson suspension. *Computers in Mechanical Engineering*, 7(2):52–56, 1988.

[6] J.P. Merlet. Kinematic analysis of suspension mechanisms. In *Proceedings of the 29th International Symposium on Automotive Technology and Automation*, Florence, 1996.

[7] C.H. Suh and C.W. Radcliffe. *Kinematics and Mechanism Design.* John Wiley and Sons, New York, 1978.

[8] K.H. Hunt. *Kinematic geometry of mechanisms.* Clarendon Press, Oxford, 1978.

[9] M. Raghavan and B. Roth. Solving polynomial systems for the the kinematic analysis of mechanisms and robot manipulators. *ASME J. of Mechanical Design*, 117(2):71–79, 1995.

[10] I.Z. Emiris and B. Mourrain. Matrices in elimination theory. *Journal of Symbolic Computation*, 28:3–44, 1999.

[11] R.E. Moore. *Methods and Applications of Interval Analysis.* SIAM Studies in Applied Mathematics, 1979.

[12] R.A. Tapia. The Kantorovitch theorem for Newton's method. *American Mathematic Monthly*, 78(1.ea):389–392, 1971.

[13] K. Yamamura, H. Kawata, and A. Tokue. Interval solution of nonlinear equations using linear programming. *BIT*, 38(1):186–199, 1998.

[14] Maple. Waterloo Maple Software Inc. http://www.maplesoft.com

[15] J.-C. Faugère. *GB Reference Manual* LITP, 1995.

[16] J.-C. Faugère and F. Rouiller. Mupad interface for GB/realsolving http://www.loria.fr/ rouillie/Soft/RS/www/index_2_3_0_0_0.html

[17] David Eisenbud, Daniel R. Grayson, Michael E. Stillman, and Bernd Sturmfels. *Computations in algebraic geometry with Macaulay 2* Algorithms and Computations in Mathematics Springer-Verlag, September 2001.

[18] M. Raghavan. An atlas of linkages for independent suspensions. *SAE Papers*, (911925), 1991.

[19] J-P. Merlet. *ALIAS: an interval analysis based library for solving and analyzing system of equations.* In SEA, Toulouse, 14-16 Juin 2000